LBNL-40110
IO-403

Automation in Construction 8 (1999) 339-350. The Netherlands. Elsevier Science B.V. 1999.

Product modeling for computer-aided decision-making

K. Papamichael, H. Chauvet, J. LaPorta and R. Dandridge

Building Technologies Department
Environmental Energy Technologies Division
Lawrence Berkeley National Laboratory
University of California
1 Cyclotron Road
Berkeley, CA 94720 USA

March 1997

This work was funded, in part, by the California Institute for Energy Efficiency (CIEE), a research unit of the University of California. Publication of research results does not imply CIEE endorsement of or agreement with these findings, nor that of any CIEE sponsor. This work was also supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Systems of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098.

Product modeling for computer-aided decision-making

K. Papamichael, H. Chauvet, J. LaPorta and R. Dandridge

Building Technologies Department, Environmental Energy Technologies Division,
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Abstract

In this paper we describe the product modeling techniques that we use for the development of a computer-aided decision-making tool for the building industry. We start with an introduction to modeling and a brief description of the goals and scope of the project, and follow with an extensive presentation and discussion of the modeling techniques employed. We conclude with a brief description of our plans for the future.

Introduction

We use the term "product" to refer to building components and systems. While the primary focus of our efforts will serve the building industry, we believe that our theories and techniques will be of value to other industries as well. The product modeling techniques described herein are being applied in the development of the Building Design Advisor (BDA), a computer-based tool that will assist decision-makers in the building industry.

Modeling theory

We understand product modeling as the representation of a product in terms of parameters that reflect its descriptive and performance characteristics. Descriptive parameters, such as geometry, color, etc., are defined herein as those controlled by the decision-maker. Performance parameters, such as comfort levels, energy requirements, etc., are defined as those that the decision-maker uses to judge the appropriateness of the product. Context parameters are those used to describe the environment within which the product is assumed and evaluated. The values of performance parameters may depend not only on the values of descriptive parameters, but context ones as well. Modeling based on these parameters facilitates communication and supports testing applications of new and existing products.

Based on the above definitions, most activities in building design are forms of modeling. Currently, the most common models used in the building industry are drawings, such as plans, sections, elevations, isometrics, perspectives, etc., as well as physical scale models. These models adequately support the evaluation of spatial layout and aesthetic appeal and are usually complimented by mathematical models that address other aspects, such as structural, energy and economic performance.

Computer-based models

Advances in computer applications over the last few decades have resulted in the gradual replacement of manual modeling with computer-simulation models. While computer-based models have been developed for a large variety of building performance considerations, computer-aided drafting models have been the most widely used in the building industry. Most others tend to be used mainly for research purposes, modeling performance aspects such as comfort, energy, and economics. Some of these models are able to address not only the building design needs, but construction and operational requirements as well.

Computer-aided drafting was originally developed to serve the needs of electronic circuitry design and typically generated very complex two-dimensional drawings. The same types of algorithms were later adapted for general drafting applications, including architectural and engineering drawings of buildings, their components and systems. The main limitation of the widely used drawing-based models is the distance between the very abstract, two-dimensional representations and the actual products the drawings suggest for construction. The major advancement in computer-graphics that facilitates the representation of three-dimensional solids and the tools to create and visualize these objects brings us one step closer to representing the actual components of construction.

Parallel to the developments in computer graphics, a large number of computer-based models, or simulations, are being developed by building researchers, that address various aspects of building performance, such as comfort, energy, economics, etc. The development of such models over the past twenty years has been broad, with various levels of success in modeling capabilities and prediction accuracy. While most models were originally implemented on mainframe and mini computers, those that are still under development have shifted their development onto powerful workstations and personal computers. Developed primarily for research purposes only, most of these applications tend to be difficult to use. They require an extensive description of the building and its context and they provide output in the form of alphanumeric tables that are cumbersome to review and interpret.

Research efforts in computer applications in the building industry during the last decade have focused on developing new models that will combine the capabilities of a large variety of existing models. These new models will provide for more cost-effective performance prediction of multiple design alternatives. In this paper we describe the results of such efforts within the Building Technologies Program of the Environmental Energy Technologies Division at Lawrence Berkeley National Laboratory.

Background

Responding to the energy crises of the early 1970s, Lawrence Berkeley National Laboratory began development of cost-effective and environmentally friendly strategies and technologies to improve the energy efficiency of buildings without compromising comfort. The research processes followed in this development are conceptually identical to building design. The main difference, however, is that research projects devote many months and even years focusing on a specific subject, while actual building projects may only be able to afford a few hours or perhaps days in consideration of the same issues. Another difference is the context within which researchers test their ideas. To understand the general performance trends of energy efficient strategies and technologies, researchers must examine them in various applications, parametrically changing key design and context parameters while keeping most constant. As a result, research findings are usually general and may not be applicable to specific applications.

Simulation tools

Extensive research efforts during the 1970s and 1980s resulted in the development of several computer-based models used to simulate building performance with respect to energy, comfort, environmental impact, economics, etc. A computer-based simulation model can be seen as the representation of and interaction among the parameters that are required to describe a phenomenon. Depending on the performance aspect being considered, these parameters and interactions may vary drastically. Walls could be “thermal barriers with U-values and areas” for thermal computations, “polygons with reflectance values and textures” for lighting computations,” or “assemblies with construction, maintenance and repair costs” for economic computations. These different modeling requirements fostered the development of independent simulation programs such as DOE-2 for energy and energy costs computations [1], SUPERLITE for daylighting computations [2], RADIANCE for lighting and rendering computations [3], COMIS for airflow and indoor air quality computations [4].

These types of simulation programs were developed over long periods of time and most of them are still under development, improving their modeling capabilities and prediction accuracy. They have proven most instrumental for the development of a large variety of energy-efficient strategies and technologies. When we initiated efforts to transfer these strategies and technologies to the building industry, we realized that the general statements about their performance were not adequate for decision-making in specific projects. Since our simulation models were very hard to use by architects and engineers, our efforts were redirected to making them easy to use routinely in everyday building design. In collaboration with several academic and research institutions, we spent several years exploring the design and decision-making process. By 1991 we had developed a design theory [5, 6] that served as the foundation for the development of a demonstration prototype that incorporated multiple simulation tools during the building design process. This tool attracted the interest of California Utilities, which initiated support for the development of the Building Design Advisor [7] through the California Institute for Energy Efficiency.

Objectives and strategies

The main objective of the Building Design Advisor (BDA) project is to develop a computer-based tool that allows building decision-makers to quickly and easily integrate energy considerations into decision-making, throughout the early phases of building design. The functional requirements of the BDA include the use of a graphic editor for the specification of the geometric attributes of building components and systems.

From our research in design theories and methods we realize that a successful computer-based prototype system must support the use of multiple simulation tools. This system must also support the various building representations required by the different simulation tools. To meet this requirement, we developed a single building model that is a superset of the parameters used by individual tools. This single model could be used to communicate with the user and is mapped to the individual representations of the simulation tools so that we can automate the preparation of their input, as well as integrate their output for multi-criterion decision making.

Another major challenge rises from the need to use detailed simulation models during the initial, schematic phases of building design, when decisions on detailed issues have not yet been made. Since the simulation tools that are linked to the BDA require values for all input parameters, we developed a schema that assigns “smart” default values to all parameters that are not yet defined by the user. Since default values represent design decisions, they are clearly indicated to the user, and may be modified at any time during the design process.

Decision-making

Our research efforts also indicated that in order to support decision-making we must not only provide a means for performance

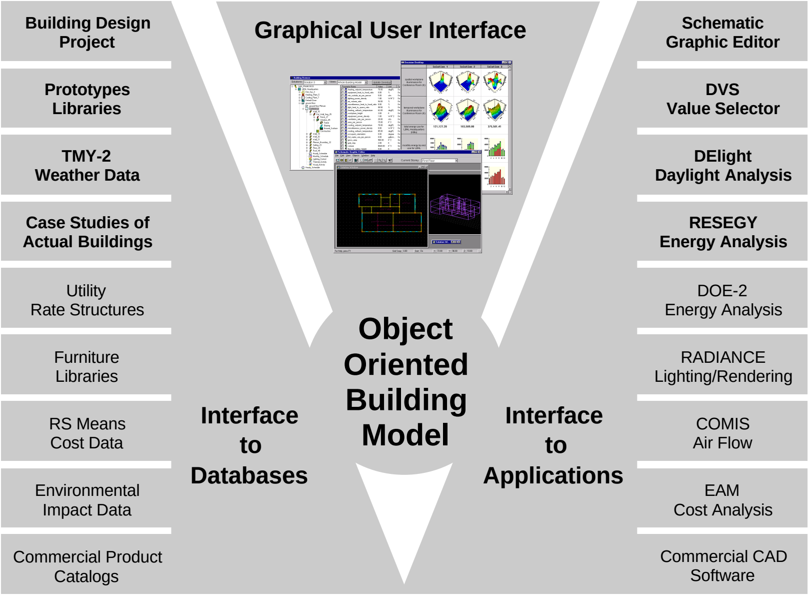


Fig 1. Schematic diagram of the Building Design Advisor general architecture, illustrating the basic strategy of a single, object-oriented model of the building and its context. The intent is to provide a single user interface for controlling the various simulation tools and databases.

prediction but for performance evaluation as well. Since performance evaluation requires comparison among alternatives, we support the evaluation of concurrent design solutions, as well as links to a Case Studies Database of actual buildings. Finally, we developed a graphical user interface that consists of two main elements: the Building Browser and the Decision Desktop [7].

The Building Browser allows building designers to quickly navigate through the multitude of descriptive and performance parameters required by the simulation tools linked to the BDA. Through the Building Browser, the user can edit the values of input parameters and select any number of parameters for display in the Decision Desktop. The Decision Desktop allows multi-criterion decision-making, through comparison of multiple alternative design solutions with respect to multiple performance parameters. The Decision Desktop supports a variety of data types, including 2-D and 3-D distributions, images, sound and video.[1]

Modeling techniques

Our overall strategy is to develop an expandable environment that supports the mapping of a single model of the building and its context to multiple simulation tools and databases, driven by a simple graphical user interface (**Fig. 1**). Following the general trends in the current approaches to representing buildings, we use an object-oriented representation of the building and its context. Since we did not know which tools we would eventually link to the BDA, we developed a model that would allow us to expand the single, object-oriented representation of the building and its context as required for the addition of simulation tools and databases in the future. This model consists of three databases and various applications that operate on them (**Fig. 2**).

[1] In this paper we focus on the modeling methods and techniques used for the development of the BDA, presenting applications that are meant for developers, rather than users of the BDA. A detailed description of the BDA application from the user's point of view, including screen captures of the BDA user interface and the Schematic Graphic Editor, is presented in reference #7.

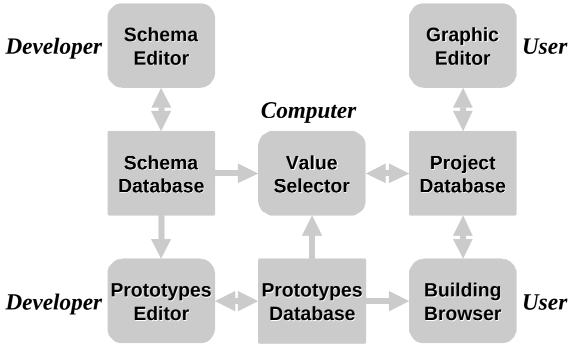


Fig. 2. Schematic diagram showing the three main databases used in the BDA and the main processes that operate on them.

The schema database

The Schema Database is a data dictionary where definitions for Building Object Types (e.g. space, wall), Properties (e.g., height, U-value), Units (e.g., ft., cm., degrees), Relations (e.g., has, faces) and Simulation Tools (e.g. DOE-2, RADIANCE, etc.) are stored. Parameters of building components and systems are then defined as links between Object Types and Properties (e.g., space height, wall U-value). Each parameter is also linked to the simulation tools that use it as input or output along with the associated type of units (**Fig. 3**).

To facilitate the development of the Schema Database we developed a Graphical User Interface that allows developers to define new Simulation Tools, Building Object Types, Properties, Units, and Relations, as well as Relationships and Parameter Lists for each Building Object Type (**Figs. 4 and 5**). The Schema Database also has reporting utilities that allow developers to check the consistency and semantics of the Schema (e.g., parameter definitions and links to simulation tools).

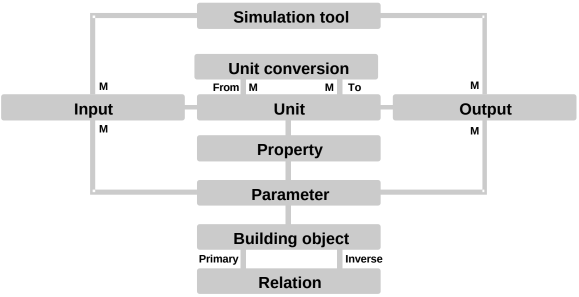


Fig. 3. Schematic diagram of the meta-schema used for the development of the Building Design Advisor, showing the main objects and their relationships.

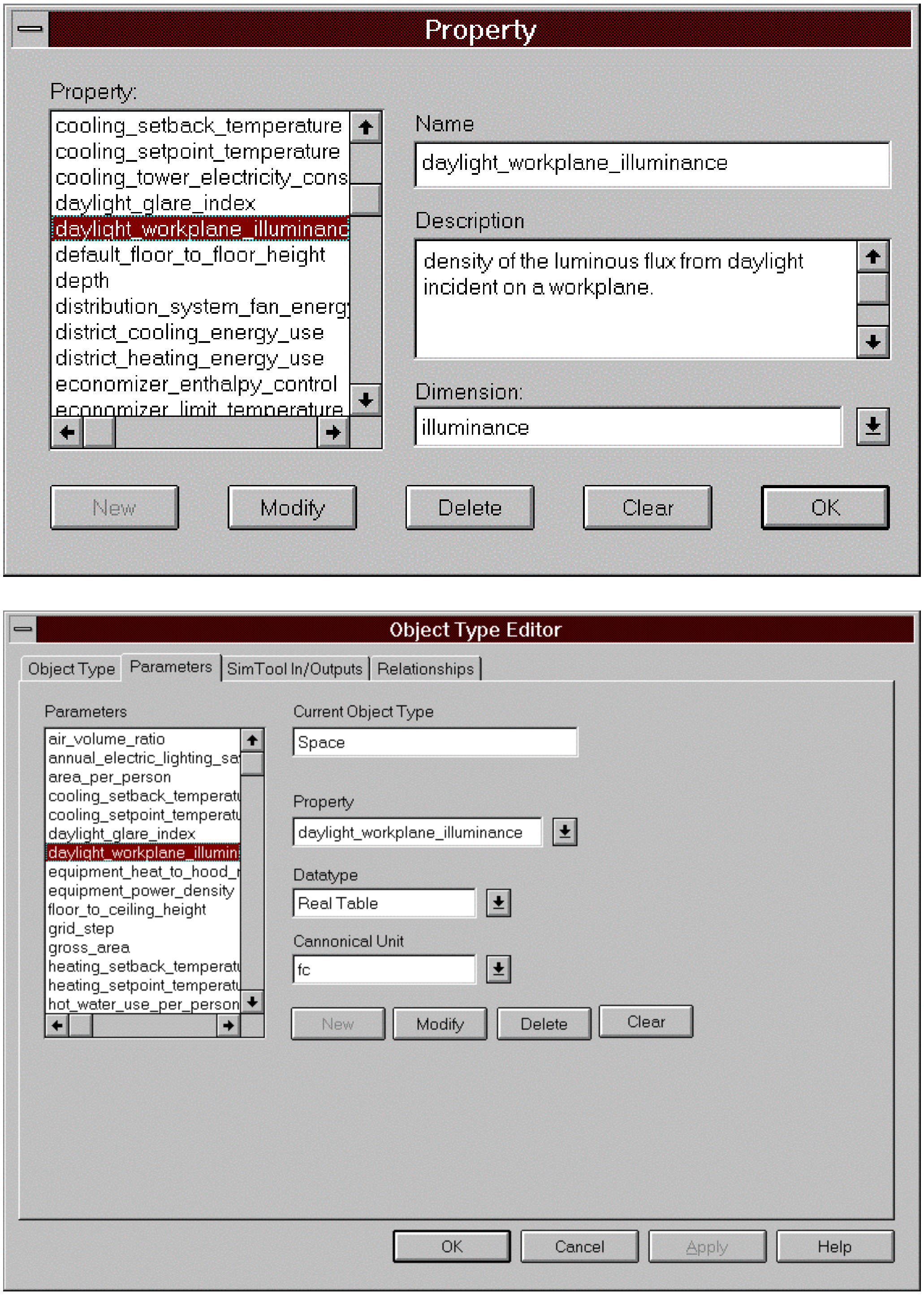
The prototypes database

The Prototypes Database is used to store Libraries of Building Object Type Instances (or Prototypes). Each Prototype is created with its own list of parameters as defined in the Schema Database, and each parameter is assigned a Value from some Source or Data Reference. The Prototypes Database is the main source of building components and systems available to the user for the description of the building. Like the Schema Database, the Prototypes Database has its own Graphical User Interface that allows developers to enter new Prototypes and modify existing ones (**Fig. 6**). Moreover, it too has reporting utilities that support the listing and printing of all Instances for each Object Type.

The project database

The Project Database is used to store the Building Object Type Instances created at run-time by the BDA. Staying with our "generic" approach, we did not define classes for different building objects. Rather, we defined classes for Run-time Building Object Type Instances, Run-time Parameter Instances, and Run-time Value Instances along with five derived classes to handle integer, real, string, real array, image, and multi-media data types.

In the BDA run-time system, the Building Object Type Instances as well as Parameters and Values are represented as C++ objects. This allows Parameters to



Property:

cooling_setback_temperature
cooling_setpoint_temperature
cooling_tower_electricity_cons
daylight_glare_index
daylight_workplane_illuminanc
default_floor_to_floor_height
depth
distribution_system_fan_energ
district_cooling_energy_use
district_heating_energy_use
economizer_enthalpy_control
economizer_limit_temperature

Name: daylight_workplane_illuminance

Description: density of the luminous flux from daylight incident on a workplane.

Dimension: illuminance

Parameters

air_volume_ratio
annual_electric_lighting_sa
area_per_person
cooling_setback_temperatu
cooling_setpoint_temperatu
daylight_glare_index
daylight_workplane_illumin
equipment_heat_to_hood_
equipment_power_density
floor_to_ceiling_height
grid_step
gross_area
heating_setback_temperatu
heating_setpoint_temperatu
hot_water_use_per_person

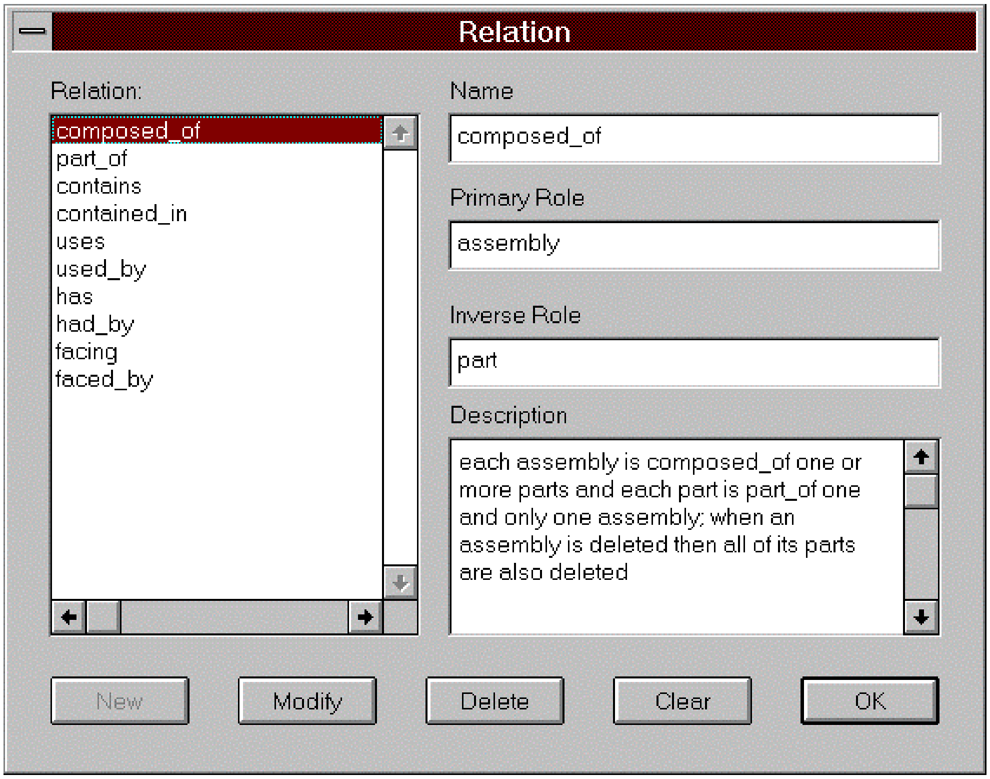
Current Object Type: Space

Property: daylight_workplane_illuminance

Datatype: Real Table

Cannonical Unit: fc

Fig. 4. Example screen captures from the Schema Editor illustrating the definition of properties and their assignment to objects as parameters.



Relation:

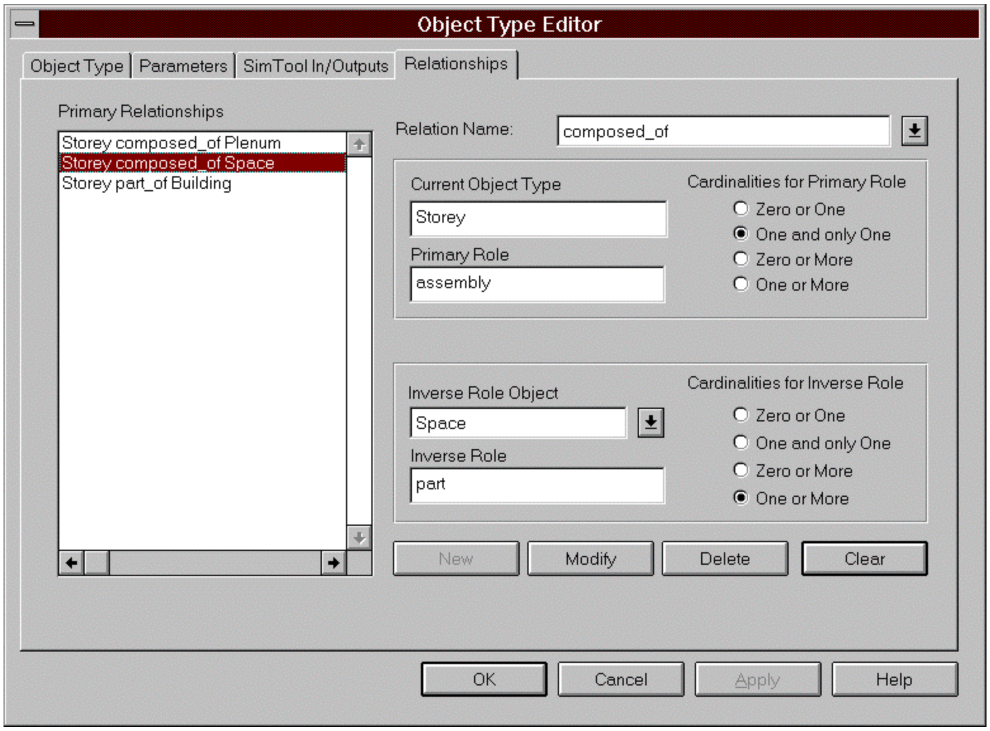
composed_of
part_of
contains
contained_in
uses
used_by
has
had_by
facing
faced_by

Name: composed_of

Primary Role: assembly

Inverse Role: part

Description: each assembly is composed_of one or more parts and each part is part_of one and only one assembly; when an assembly is deleted then all of its parts are also deleted



Primary Relationships

Storey composed_of Plenum
Storey composed_of Space
Storey part_of Building

Relation Name: composed_of

Current Object Type: Storey

Primary Role: assembly

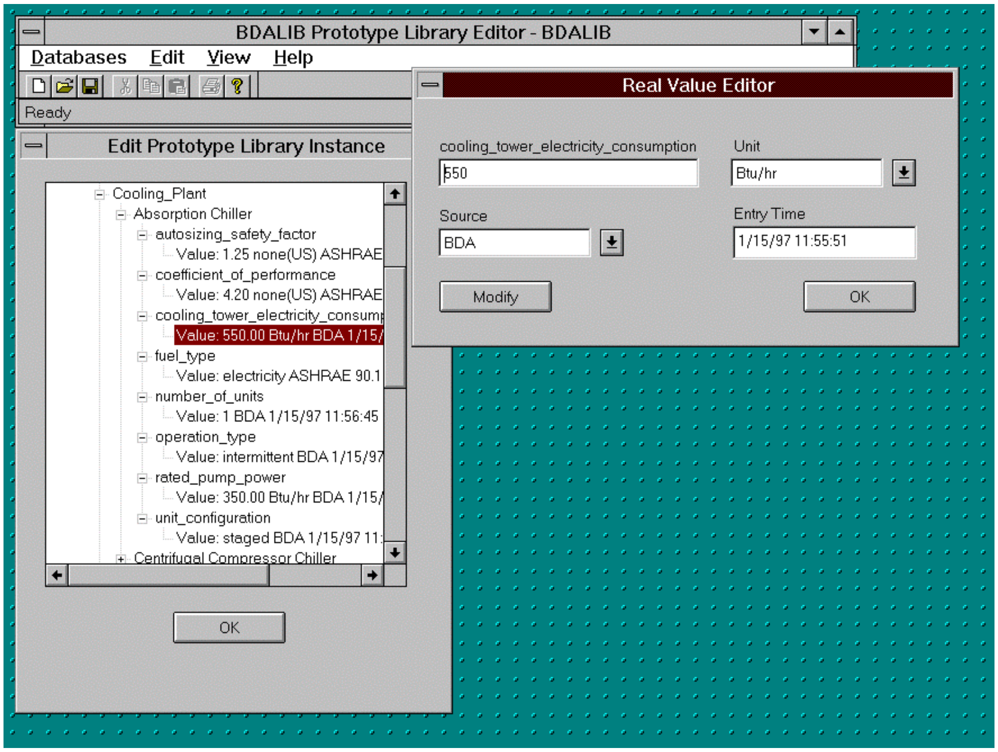
Cardinalities for Primary Role
○ Zero or One
● One and only One
○ Zero or More
○ One or More

Inverse Role Object: Space

Inverse Role: part

Cardinalities for Inverse Role
○ Zero or One
○ One and only One
○ Zero or More
● One or More

Fig. 5. Example screen captures from the Schema Editor illustrating the definition of relations and their use in defining relationships between pairs of objects.


BDALIB Prototype Library Editor - BDALIB
Databases
Edit
View
Help
Ready
Edit Prototype Library Instance
Real Value Editor
Modify
OK
OK

```
Cooling_Plant
  Absorption Chiller
    autosizing_safety_factor
      Value: 1.25 none(US) ASHRAE
    coefficient_of_performance
      Value: 4.20 none(US) ASHRAE
    cooling_tower_electricity_consump
      Value: 550.00 Btu/hr BDA 1/15/
    fuel_type
      Value: electricity ASHRAE 90.1
    number_of_units
      Value: 1 BDA 1/15/97 11:56:45
    operation_type
      Value: intermittent BDA 1/15/97
    rated_pump_power
      Value: 350.00 Btu/hr BDA 1/15/
    unit_configuration
      Value: staged BDA 1/15/97 11:
  Centrifugal Compressor Chiller
```

cooling_tower_electricity_consumption: 550

Unit: Btu/hr

Source: BDA

Entry Time: 1/15/97 11:55:51

Fig. 6. Example screen capture from the Prototypes Editor illustrating the definition of specifications for an absorption chiller.

more than one Value, each from a separate Source or Simulation Tool. The reason for this "expensive" representation is the desire to use the BDA environment for the implementation of a Building Lifecycle Information Support System (BLISS). BLISS will expand beyond original design to address the data needs of building construction, commissioning, operation, and so on. To satisfy the need for performance evaluation, the BDA supports multiple design alternatives within a project database. A new alternative design solution is generated at any point as a copy of any of the existing solutions. The BDA user interface supports the concurrent review and manipulation of any number of alternative design solutions. Moreover, it supports their side-by-side comparison with respect to multiple performance considerations.

The building model

The development of the Schema Database is guided by the tools that we are linking to the BDA. For the 1.0 version, we have been addressing the needs of DElight, a daylighting tool that uses the DOE-2 daylighting algorithms [8], RESEGY, a simplified thermal and energy analysis tool [9], and SGE, a Schematic Graphic Editor that we developed specifically for the BDA. SGE allows users to graphically enter the geometric attributes of building components and systems, thereby modeling building components as opposed to merely using lines to represent them [7].

Currently, the BDA building model is a network schema with five types of relations used to link the various building components and systems among themselves as well as with the objects used to define the building context (**Figs. 7 and 8**). All relations are defined as pairs of primary and inverse expressions as follows:

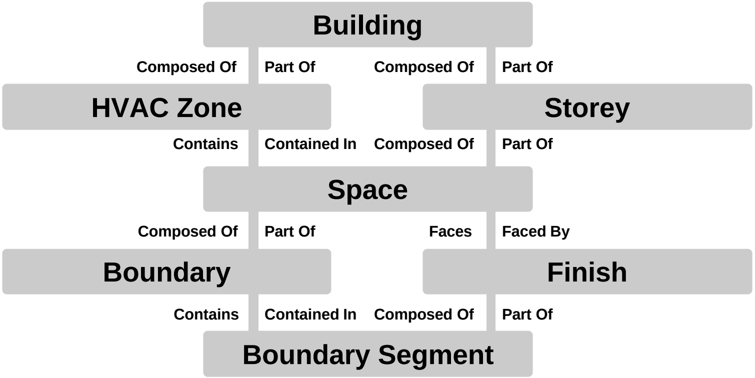


Fig. 7. Partial view of the building model focusing on the schema that relates the building to the spaces its boundaries.

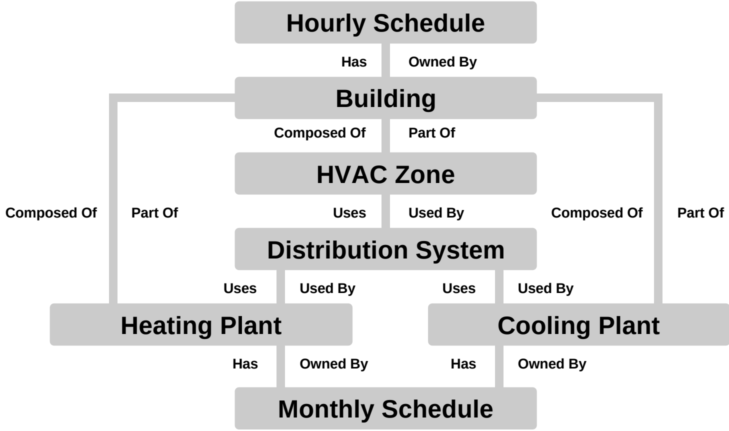


Fig. 8. Partial view of the building model focusing on the schema that relates the building to the HVAC system.

Composed Of/Part Of—An object may be an assembly that is composed of one or more parts. When an assembly is deleted, then all of its parts are also deleted. Each part is part of one and only one assembly. Deletion of a part has no effect on the existence of its assembly.

Contains/Contained In—An object may be a container, that is it may contain one or more contents. The deletion of a content has no effect on the existence of its container. Each content may be contained in one or more containers. When a container is deleted, only those contents are deleted that do not have either a part of or a contained in relationship to any other container.

Has/Owned By—An object may be an owner, that is it may have one and only one feature of a particular object type. Deletion of a feature has no effect on the existence of its owner. Each feature is owned by one and only one owner. When an owner is deleted, then all of its features are also deleted.

Uses/Used By—An object may be a client, using one or more *servers*. Client deletion has no effect on the existence of a server. Each server is used by one or more clients. Server deletion has no effect on the existence of a client but it does eliminate the service that was provided.

Faces/Faced By—This is a special relation that we use to address spaces and their boundaries. A boundary's finish faces one and only one space. Boundary deletion has no effect on the existence of the space that it faces. Each space is faced by one or more boundary finishes. When a space is deleted, then boundaries whose finishes do not face other spaces are deleted, while the ones whose finishes face other spaces may switch to a different instance (e.g., from interior to exterior wall).

Data Assignment Scenario

During the creation of a new space in the Schematic Graphic Editor, the user is asked to select a Space Prototype from those available in the Prototypes Database, such as "Lobby," "Conference Room," etc. When the "Space" Run-time Building Object Instance is created, its Object Type field is dynamically set to "Space," the Schema Database is queried for the list of Parameters required for a "Space," and Run-time Parameters are created and placed in the "Space" Run-time Object parameter list. Following this, the Prototypes Database is queried and Run-time Values are created for each Run-time Parameter in the parameter list of the "Space" Run-time Object. The rules for the selection of the default values follow building codes, standards, and recommended practice. These values are drawn from a number of sources, such as the ASHRAE Handbook of Fundamentals [10] or the Handbook of the Illuminating Engineering Society [11]. The user may change the default values at any point through the BDA user interface.

Discussion

The greatest difficulty in incorporating simulation tools within a building design system is that the CAD system and the simulation tools are radically different. The primary goal of a CAD system is to allow the user to specify and manipulate geometry. This is accomplished by representations utilizing various graphic objects (entities or symbols), such as lines and polygons. These entities can be easily created and manipulated by the user because as an object, a CAD polygon knows how to display itself, show grip handles at its vertices, and respond to mouse clicks and drag events. The second model is that of the physical world. In the A/E/C industry, this is a model of building objects such as spaces, walls, windows, etc. The Physical model is rich with non-geometric attributes, but does not translate readily into a graphic version on a computer screen. This Physical model required by simulation engines that reason about various domain parameters in energy, comfort, structures, etc. A simple example of the disparity between the two models can be seen in a wall object. In the Physical model, a wall object contains a long list of non-geometric attributes such as surface reflectance, materials, U-Values, and a set of relationships to other objects such as spaces, doors and windows. The wall would contain only that geometry necessary to describe itself in the real world—that is a list of vertices. By contrast, in the CAD model, the Wall object consists of a polygon defining the wall, layer information, lines styles, color, pen thickness, and rich set of methods which allow it to display itself and be modified through mouse-based interactions with the user.

The Dual Model Approach

In the BDA, the disparate needs of the two models resulted in two separate applications to create and maintain them. The CAD model is maintained by the Schematic Graphic Editor (SGE) and the Physical model is maintained by the BDA in the Project Database. Because the SGE is a separate application built on top of a third party library of CAD functions with no interface to our database management system, all the objects drawn in SGE are saved to a file that is independent of the BDA Project database. As a result, the two separate representations must be synchronized during the "save" and "load" operations. This synchronization problem prevents us from using the Project database to its full advantage. If both the SGE and the BDA operated on the same Building Object, then changes could be saved as they occur and only those parts of the Project database that needed to be displayed would be loaded, truly utilizing all of the advantages of a Data Base Management System. The only advantage of our current approach is that the Physical model in the BDA can be kept free of the large amounts of CAD information that is extraneous to the needs of the simulation tools.

The single, integrated model

The most viable long-term solution that we see is the merging of the two models into one that supports both the CAD functionality required by the user, and the database functionality required by the simulation tools. In this approach, the wall object will "know" everything about being a wall in the physical world, as well as how to display itself on the screen and respond to mouse clicks and drags. Unfortunately, such an environment does not yet exist. However, the industry is moving closer to it with the efforts of the International Alliance for Interoperability (IAI). The IAI is developing the Industry Foundation Classes (IFC), which is a data model that will encompass both the graphic needs of CAD systems and the data needs of analysis tools. The other goal of the IAI is that of standardization. If a standardized model existed, then the conversions between different CAD and analysis programs would be eliminated. No conversion would be required since all programs would simply create IFC Wall objects, IFC Window objects, etc. However, a new generation of simulation tools may have to be written to take full advantage of this approach.

Real versus Conceptual Objects

Another major challenge in our development efforts has been the modeling of conceptual objects, such as plenums, schedules, activities, etc., which do not really exist as "real," physical objects. The most common and problematic conceptual object is the "space." The space has been a focal point not only in the required functionality of the Schematic Graphic Editor (e.g., users want to be able to "move spaces around…") but in the modeling of the simulation tools as well. Most daylight calculations are performed on a space-by-space basis, as are many of the thermal and air quality calculations. This is intrinsically problematic in modeling because the object that we consider as most important does not exit in the physical realm. The space is an abstraction that permits us to reason about a given volume that is defined by a combination of physical and imaginary boundaries. In the simplest case, a space is defined on all sides by physical boundaries (e.g., walls, floor and ceiling). However, spaces can also be defined by a small change in elevation, or a change in the floor material, or by completely imaginary boundaries that we use to mentally "close" a room, but which do not exist in the physical world.

Boundaries and Boundary Segments

The approach that we have taken in the BDA is to allow the user to define each space by drawing a polygon in the SGE, explicitly closing it. Then, after the space has been defined, the user may edit specific space boundaries and designate their construction to NULL. This provides for an exact definition of the space, while allowing for non-physical boundaries. One problem that arises from this approach is that walls shared by two spaces are defined twice, since each space is explicitly described. To solve this problem we introduced the notion of the Wall Segment object. While Wall objects are still used to define the perimeter of each space, each Wall is composed of one or more Wall-Segments. When the SGE detects overlapping Walls, these are automatically segmented into the proper number of Wall Segments, so that there is no overlap. The Wall Segment is then used to define the construction and other physical attributes required by the simulation tools. Through this approach we model the conceptual boundaries of the space using Wall objects and the physical boundaries of the space using Wall-Segments. The automatic generation and maintenance of the Wall-Segment objects has been one of the most challenging implementation efforts of the SGE. This functionality allows the user to freely move entire spaces at any time during the design process, which is extremely important during the early, schematic phases of building design.

Conclusions

Our product modeling efforts in the development of the Building Design Advisor have been directed towards an expandable system that will potentially satisfy the needs of many simulation tools and databases. In the current building model we have implemented the objects and parameters needed for DElight and RESEGY. During this implementation, however, we have been addressing and considering the data needs of linking to additional tools that we plan to implement in the future.

For the 2.0 version of the BDA we plan to develop links to DOE-2 and RADIANCE. Our plans for future expansions include the development of links to COMIS, as well as economic analysis and environmental impact tools, building rating tools, commercial CAD software and electronic catalogs of actual products from manufacturers of building components and systems. We are also participating in the International Alliance for Interoperability efforts and plan to implement the Industry Foundation Classes when they will reach a level that satisfies the data needs of the tools that are linked to the BDA.

Acknowledgments

This work was funded, in part, by the California Institute for Energy Efficiency (CIEE), a research unit of the University of California. Publication of research results does not imply CIEE endorsement of or agreement with these findings, nor that of any CIEE sponsor. This work was also supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Systems of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098.

References

[1] F.C. Winkelmann, B.E. Birdsall, W.F. Buhl, K.L. Ellington, A.E. Erdem, J.J. Hirsch and S.D. Gates: “DOE-2 Supplement: Version 2.1E” Lawrence Berkeley Laboratory report no. LBL-34947, 1993.

[2] M. Modest: “A general model for the calculation of daylighting in interior spaces,” *Energy and Buildings*, Vol. 5, pp. 66-79.

[3] G. J. Ward: “Visualization.” Lighting Design and Application, Vol. 20, No. 6, pp. 4-20, 1992.

[4] H.E Feustel: “Annex 23 multizone airflow modeling – an international effort,” Proceedings of the International Symposium on Air Flow in Multizone Structures, Budapest, Hungary, 1992.

[5] K. Papamichael: “Design process and knowledge; possibilities and limitations

of computer-aided design." *Ph.D. Dissertation*, Department of Architecture, University of California, Berkeley, CA, August 1991.

[6] K. Papamichael and J.P. Protzen: "The Limits of Intelligence in Design," Proceedings of the Focus Symposium on Computer-Assisted Building Design Systems, of the Fourth International Symposium on System Research, Informatics and Cybernetics, Baden-Baden, Germany, August 3-4, 1993.

[7] K. Papamichael, J. LaPorta, H. Chauvet: "Building Design Advisor: automated integration of multiple simulation tools," *Automation in Construction*, Vol. 6, No. 4, August 1997.

[8] F.C. Winkelmann: "Daylighting calculation in DOE-2," LBNL report No. LBL-11353, III.2.9, Lawrence Berkeley National Laboratory, 1983.

[9] W.L. Carroll, B.E. Birdsall, R.J. Hitchcock, and R.C. Kammerud: "RESEM: An evaluation tool for energy retrofits in institutional buildings," Proceedings of the International Building Performance Simulation Association, 1989, pp. 107-112.

[10] ASHRAE: *ASHRAE Fundamentals Handbook*, American Society of Heating, Refrigerating and Air-conditioning Engineers, 1993.

[11] IESNA: *IESNA Handbook*, Illuminating Engineering Society of North America, 1993.